

Selected Flight Test Results for Online Learning Neural Network-Based Flight Control System*

Peggy S. Williams[†]

NASA Dryden Flight Research Center, Edwards, CA, 93523, USA

The NASA F-15 Intelligent Flight Control System project team has developed a series of flight control concepts designed to demonstrate the benefits of a neural network-based adaptive controller. The objective of the team is to develop and flight-test control systems that use neural network technology to optimize the performance of the aircraft under nominal conditions as well as stabilize the aircraft under failure conditions. Failure conditions include locked or failed control surfaces as well as unforeseen damage that might occur to the aircraft in flight. This report presents flight-test results for an adaptive controller using stability and control derivative values from an online learning neural network. A dynamic cell structure neural network is used in conjunction with a real-time parameter identification algorithm to estimate aerodynamic stability and control derivative increments to the baseline aerodynamic derivatives in flight. This set of open-loop flight tests was performed in preparation for a future phase of flights in which the learning neural network and parameter identification algorithm output would provide the flight controller with aerodynamic stability and control derivative updates in near real time. Two flight maneuvers are analyzed – a pitch frequency sweep and an automated flight-test maneuver designed to optimally excite the parameter identification algorithm in all axes. Frequency responses generated from flight data are compared to those obtained from nonlinear simulation runs. An examination of flight data shows that addition of the flight-identified aerodynamic derivative increments into the simulation improved the pitch handling qualities of the aircraft.

Nomenclature

ACH	=	average Cooper-Harper
CAP	=	control anticipation parameter
DCS	=	dynamic cell structure
Gen 1	=	generation 1
IFCS	=	Intelligent Flight Control System
LQR	=	linear quadratic regulator
n_z	=	normal acceleration, g
PID	=	parameter identification
PTNN	=	pretrained neural network
SOFFT	=	stochastic optimal feedforward and feedback technology
α	=	angle of attack, deg
β	=	angle of sideslip, deg
δ_s	=	individual stabilator deflection, deg
C_m	=	pitching moment coefficient
C_z	=	z-axis force coefficient

* The use of trademarks or names of manufacturers is for accurate reporting and does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

[†] Aerospace Engineer, Dynamics and Controls Branch, P.O. Box 273/MS-4840D, Member.

I. Introduction

The NASA F-15 Intelligent Flight Control System (IFCS) project incorporates a series of flight control concepts that are designed to demonstrate the benefits of a neural network-based adaptive controller. The primary goal of the IFCS team is development and flight testing of the control systems that use neural network technology to optimize the aircraft performance under nominal conditions in addition to the ability to stabilize the aircraft under failure conditions. These failure conditions encompass locked or failed control surfaces as well as unforeseen damage that might occur to the aircraft in flight.

The baseline configuration of the IFCS consists of a pretrained neural network (PTNN) and a model-following optimal controller, known as the stochastic optimal feedforward and feedback technology (SOFFT) controller. The PTNN provides aircraft stability and control derivatives to the SOFFT controller. The SOFFT controller uses these derivatives to stabilize the vehicle and provide desired flying characteristics. This baseline configuration of the PTNN and SOFFT controller was originally flown on the NASA 837 airplane in 1999, and then again in 2002 after the IFCS program was reinstated.

The IFCS baseline configuration was augmented with a parameter identification (PID) algorithm¹ that identifies aerodynamic stability and control derivatives in near real time during flight. An online learning neural network, dynamic cell structure (DCS),² was also added to the system with the objectives of learning the changes in the stability and control derivatives identified by the PID algorithm and of being able to recall derivatives based on flight condition. This control system in its entirety, encompassing the PTNN, SOFFT, PID, and DCS, is referred to as the generation 1 (Gen 1) IFCS. The Gen 1 configuration was flight tested in an open-loop configuration during the summer of 2003. In these open-loop flight tests, DCS aerodynamic derivative increments were not passed on to the SOFFT controller, but were recorded onboard and also telemetered to the control room on the ground.

This report documents the analysis of data obtained from the open-loop Gen 1 flight tests in regard to the pitch axis derivatives for selected in-flight maneuvers. Flight data are analyzed to determine whether the PID algorithm was able to identify either changes or inaccuracies in the baseline aerodynamic derivatives. Simulation studies are conducted to show that the PID-predicted derivatives measured in flight are accurate in the sense that they would have produced an improvement in longitudinal handling qualities had the system been flown in a closed-loop fashion.

II. Test Airplane

The test airplane, NASA 837, is a highly modified preproduction F-15B airplane (Fig. 1) and is not representative of production F-15 aircraft. Modifications to the airplane include two canards mounted on the upper inlet area forward of the wing. The canards are modified F-18 horizontal tail surfaces, and their position in flight is scheduled with angle of attack (α). An additional modification to the airplane included the incorporation of two F100-PW-229 Pratt & Whitney (West Palm Beach, Florida) engines with axisymmetric thrust vectoring nozzles. During flight test, the thrust vectoring capability was enabled, but the vectoring is commanded to zero. The airplane is controlled by a quadruplex, digital, fly-by-wire, flight control system. All mechanical linkages between the control stick, rudder pedals, and controls surfaces have been removed from the airplane.³



Figure 1. NASA 837 F-15 airplane.

III. Intelligent Flight Control System Architecture

Figure 2 shows the Gen 1 IFCS architecture. The baseline configuration consists of the PTNN and the SOFFT controller. The PTNN is a nonlearning neural network that functions as an aerodynamic derivative table lookup based on Mach number, α , angle of sideslip (β), and the control surface positions. These baseline aerodynamic derivatives are sent from the PTNN to the SOFFT controller.

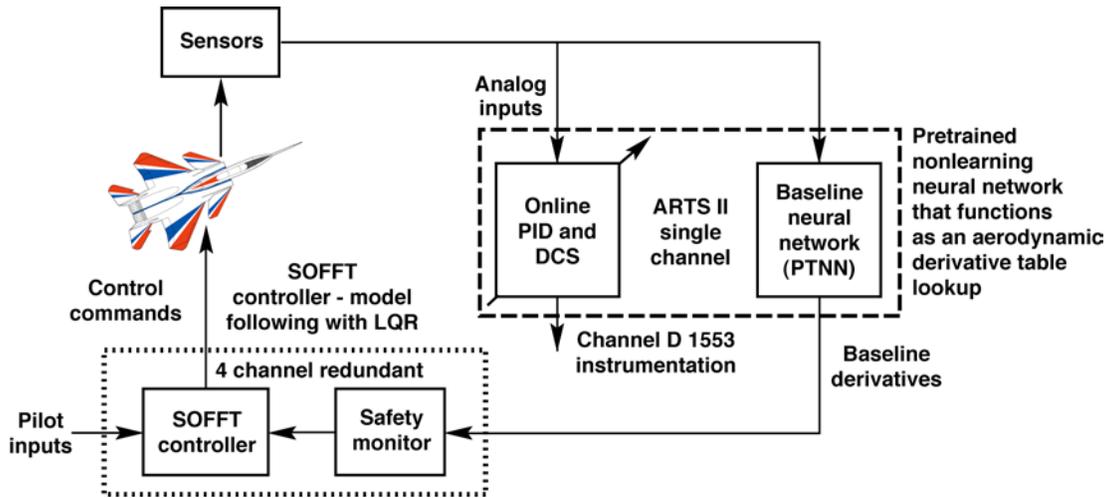


Figure 2. IFCS architecture.

A. Pretrained Neural Network

The PTNN uses a Levenberg-Marquardt neural network to model the aerodynamic information normally contained in numerous tables in the nonlinear simulation. Because the PTNN uses relatively little memory, the flight controller has access to the entire aerodynamic database of the airplane. In the original program, the PTNN was trained on aerodynamic information using the square nozzle configuration of the airplane engines instead of the current round nozzle (thrust vectoring) configuration of the engines. This mistake was not realized until after the flight tests were completed in 1999. Degraded pitch handling qualities seen in those flights were correctly attributed to the inaccurate derivative values used to train the PTNN.⁴

When the IFCS program was reinstated in 2001, it was decided that the PTNN should not be retrained. This created a “built-in” experiment to test whether or not the PID and DCS algorithms would be able to identify the error in the PTNN aerodynamic derivatives and correct for them in real time, in effect accounting for the difference between round and square nozzle aerodynamics.

B. Stochastic Optimal Feedforward and Feedback Technology Controller

The IFCS uses a model-following optimal controller called SOFFT. The SOFFT controller, developed by the Boeing Phantom Works (St. Louis, Missouri) under a NASA Langley (Langley, Virginia) contract, is a linear quadratic regulator (LQR) direct discrete controller. LQR control was chosen because it best satisfied the requirement of being implemental in real time, had good stability properties, and was suitable for readjusting flight dynamics parameters.

The SOFFT flight controller uses a feedforward model to generate tracking commands. These tracking commands are combined with an LQR optimal feedback control law that adds robustness and minimizes tracking error. The flight controller uses an online solution of the Riccati equation containing the baseline neural network stability and control derivative data to continuously optimize the feedback gains.⁴

The SOFFT controller commands airplane surface positions in an attempt to follow a reference model. A second-order model was used both for the longitudinal and directional axes and a first-order model for the lateral axis. The coefficients contained in these first- and second-order models were scheduled on flight condition. Thus,

these handling qualities models are nonlinear and change significantly during short period maneuvering of the airplane. The requirement for the SOFFT controller is to continually match or follow these explicit models.

For the SOFFT controller, handling qualities parameters such as short-period frequency and damping, roll-mode time constant, and dutch roll frequency and damping, can be varied through the use of test sets that are selectable by the pilot during flight. The handling qualities parameter values for the default research-mode setting are chosen to obtain Level 1 handling qualities.

C. Parameter Identification

The PID algorithm estimates the aerodynamic stability and control derivatives in real time during flight. Because the estimates are not always accurate, a system of confidence tests was devised.¹ These tests evaluated the PID results based on estimated variances, information content, and persistence. An overall validity flag is created for each parameter estimate. Reference 1 contains additional detailed information on these tests. The PID compares the derivative estimate with the PTNN value and sends a difference to the DCS neural network. The PID algorithm also sends a validity flag with each derivative increment estimated. The DCS neural network learns on valid PID derivative data, ignoring any data that does not pass PID validity tests.

The original PID configuration estimated 26 stability and control derivatives. Changes were made, however, to remove problems initiated by correlated control surfaces positions caused by pilot inputs. This modified version, called the tuned-down PID, only estimated 13 of the 26 stability and control derivatives (removing the aileron, canard, and angular rate derivatives). PID sends a zero correction for the removed 13 derivatives, as well as a false validity flag, in effect passing the PTNN value unmodified through the DCS to the SOFFT controller. Because the values of the nonestimated derivatives come solely from the PTNN, a possible source of error for the estimated derivatives is introduced into the system.

D. Dynamic Cell Structure

The DCS neural network was developed by NASA Ames Research Center (Moffett Field, California) and incorporated into the virtual reality simulator designed by the Ames Neuro Engineering Lab.² The DCS is a topology-representing or self-organizing map type of neural network. DCS was chosen because of the characteristics of long-term memory and the ability to change the neural network size by adding nodes and edges. Kohonen learning was used to adjust the location of the nodes, while Hebbian learning was used to adjust the strength of the edges.²

For flight, the DCS algorithm was modified into five distinct networks to improve the speed and accuracy.⁵ Various parameters were tuned to improve performance and were changeable by way of a configuration file that could be loaded prior to a particular flight. Inputs to the DCS include Mach, altitude, α , β , and surface positions, along with the derivative increments estimated by the PID and their respective validity flags. The DCS outputs were the 26 aerodynamic stability and control derivative increments.

E. Safety Monitor

A safety monitor was implemented as a safeguard and check on the aerodynamic derivative values coming from the PTNN, PID, and DCS, because the computations associated with the learning system are not processed in the flight control computer. The safety monitor uses checksums to verify whether the values computed are reasonable and within known limits. The derivative values are broken into groups and checksum values are formed using derivative values that have been normalized between 0 and 1. These checksum values are then compared to predetermined values contained in a table lookup. The IFCS will downmode if a one-time error threshold is exceeded or if a persistence error threshold is exceeded for a certain number of timeframes. The safety monitor ensures that the controller will not receive invalid values for the stability and control derivatives during flight, and furthermore helps ensure safety of flight.

IV. Results

An automated maneuver⁶ was performed at the flight condition of Mach 0.75 at an altitude of 20,000 ft. This maneuver, which lasts for 15 seconds, moves the various control surfaces at independent frequencies. The automated maneuver provides excitation for the PID algorithm that enables the PID to estimate all stability and control derivatives simultaneously. Figure 3 shows surface positions for one of these automated maneuvers performed in flight.

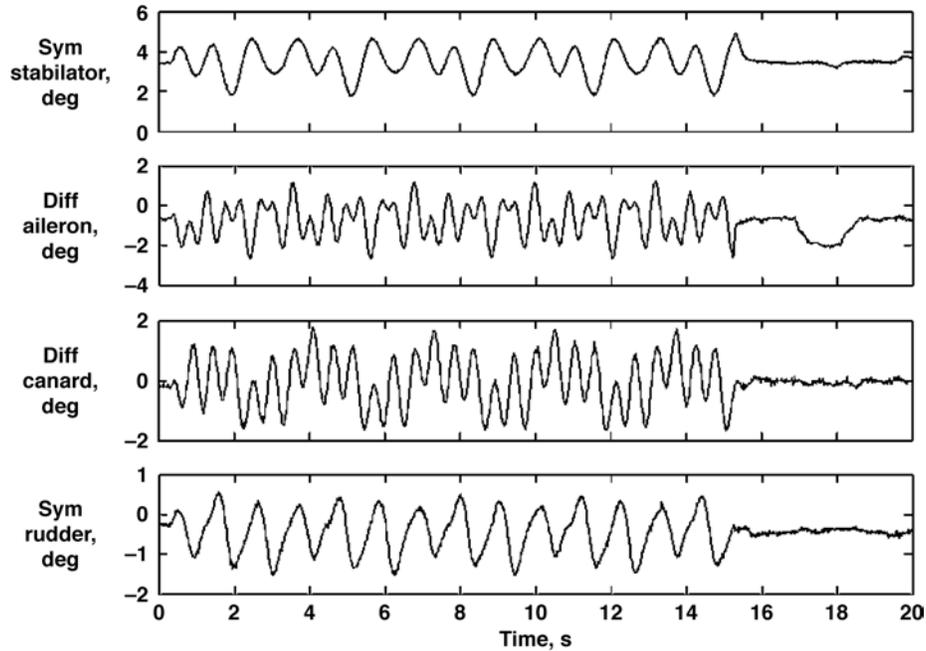


Figure 3. Surface deflections for automated PID excitation maneuver.

Because the round as opposed to the square nozzle configuration of the airplane was believed to have the most effect on the pitching moment, the behavior of the aerodynamic derivatives of pitching moment as a result of angle of attack (C_{m_α}) and pitch moment as a result of symmetric stabilator deflection ($C_{m_{\delta_s}}$) were examined. Figure 4 shows the value of the $C_{m_{\delta_s}}$ derivative during the automated maneuver. The solid blue line shows the value of the PTNN baseline derivative. The dashed green line shows the PID estimate – with the green asterisks indicating where the confidence tests declared the derivative estimate to be valid. The red dashed line shows the DCS value of the $C_{m_{\delta_s}}$ derivative. The DCS total derivative estimate matches the PTNN value until the first PID validity flag is sent at approximately 4 seconds. The DCS tends to smooth out the PID derivative estimate. This figure illustrates that the PID and DCS algorithms working together were able to identify a change to the baseline (PTNN) $C_{m_{\delta_s}}$ derivative in real time during flight.

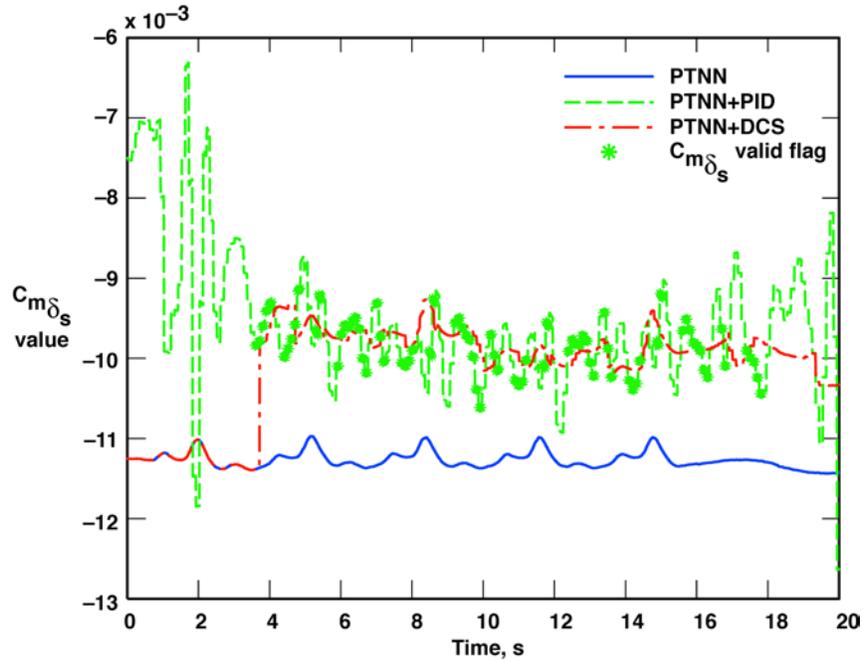


Figure 4. Comparison of $C_{m_{\delta_s}}$ aerodynamic derivative values.

The automated maneuvers were examined in an effort to determine whether the PID estimated consistent values for the aerodynamic derivative increments (Fig. 5). Repeatability was also examined in order to determine a representative value for the aerodynamic derivative increments for use in the simulation studies. The flight-identified values were calculated by averaging PID estimate values that were accompanied by valid PID flags. Table 1 shows the calculated flight-identified values for each automated maneuver. For the two automated maneuvers, the $C_{m_{\alpha}}$ and $C_{m_{\delta_s}}$ flight-identified increments estimated by the PID are consistent. The flight-identified PID estimates for the $C_{z_{\alpha}}$ and $C_{z_{\delta_s}}$ derivatives were less consistent between the two automated maneuvers. Because the PID derivative estimates were less consistent for the $C_{z_{\alpha}}$ and $C_{z_{\delta_s}}$ derivatives, the PID derivative increments for the $C_{z_{\alpha}}$ and $C_{z_{\delta_s}}$ derivatives are chosen from the first automated maneuver. The changes in the $C_{m_{\delta_s}}$, $C_{z_{\alpha}}$, and $C_{z_{\delta_s}}$ derivatives are relatively small compared to the baseline PTNN aerodynamic values. The $C_{m_{\alpha}}$ derivative value represents a change of more than 50 percent to the baseline $C_{m_{\alpha}}$ value.

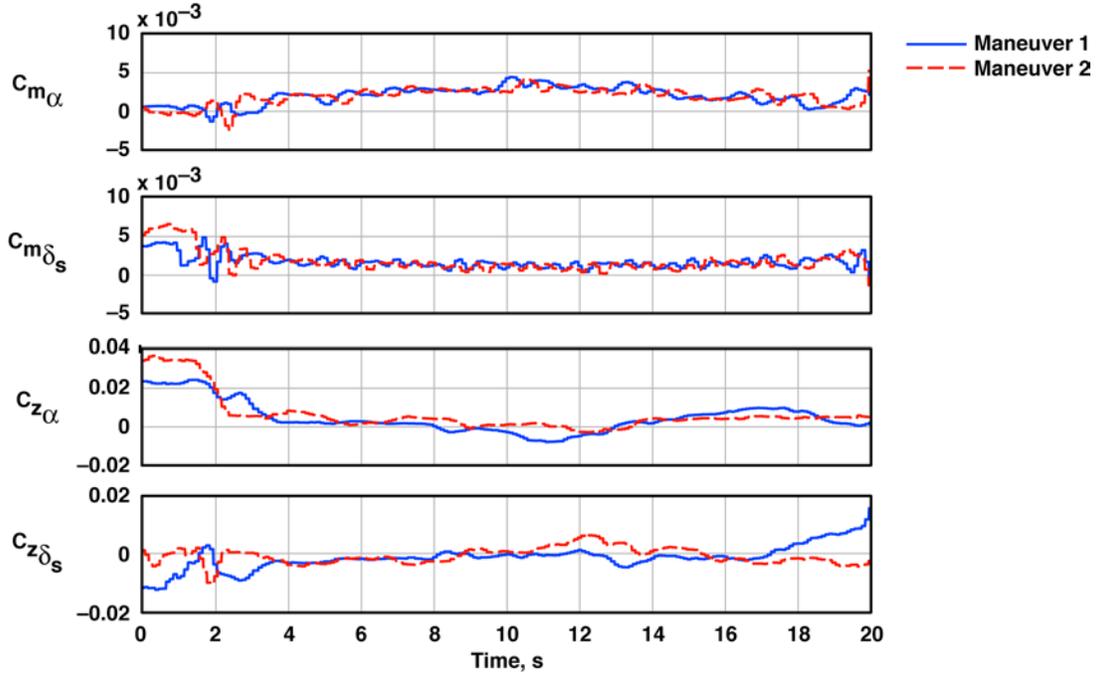


Figure 5. Incremental pitch aerodynamic derivative values estimated by the PID in flight during two automated maneuvers.

Table 1. PID-estimated increments of pitch derivatives for both automated maneuvers.

PID parameter	Automated maneuver 1 increment	Automated maneuver 2 increment
$C_{m\alpha}$	0.0025	0.0024
$C_{m\delta_s}$	0.0014	0.0013
$C_{z\alpha}$	0.0017	0.0030
$C_{z\delta_s}$	-0.0012	-0.0006

To determine whether these PID-identified increments were valid, the feedforward state space matrices were used to calculate the transfer function using the feedforward stabilator commanded deflection (δ_s) as input and normal acceleration (n_z) and pitch rate as outputs. Frequency responses were calculated for an automated pitch frequency sweep performed in the NASA Dryden Flight Research Center (Edwards, California) six-degree-of-freedom nonlinear simulation with the baseline aerodynamics in place. For comparison purposes, an automated pitch frequency sweep was run in the simulation with the incremented pitch derivative values identified in flight test being passed on to the SOFFT controller.

Figures 6 and 7 show Bode plots generated from these frequency sweeps. These Bode plots were compared to plots from a pitch frequency sweep performed by the pilot in flight. Figure 6 shows the frequency response for the stabilator deflection to n_z transfer function. Figure 7 shows the frequency response for the stabilator deflection to pitch rate transfer function. Simulation data from the instance with the PID increments identified in flight matches flight data more closely overall than simulation data with no increments added to the baseline derivatives.

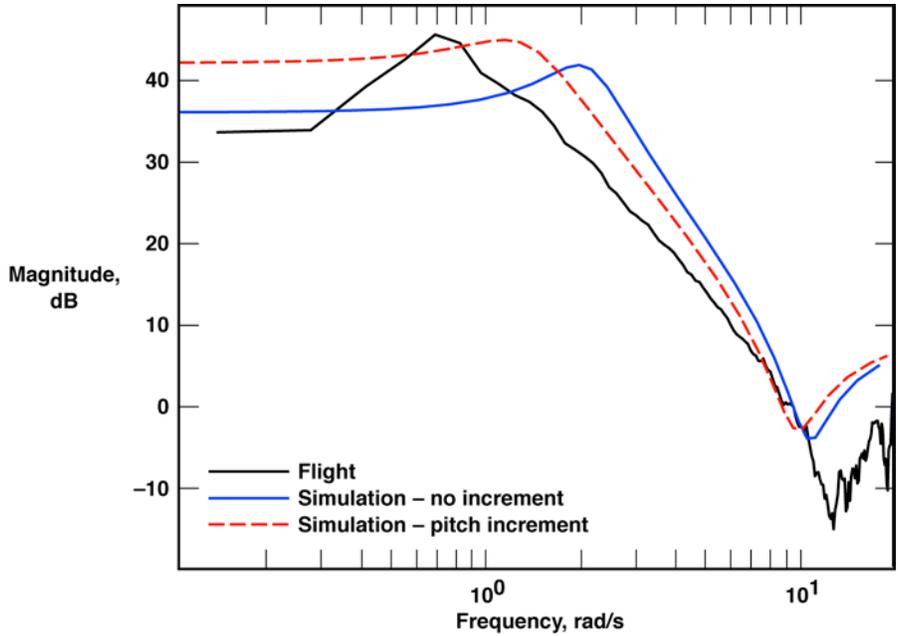


Figure 6. Bode plot comparison of flight data to simulation data – n_z to stabilator deflection.

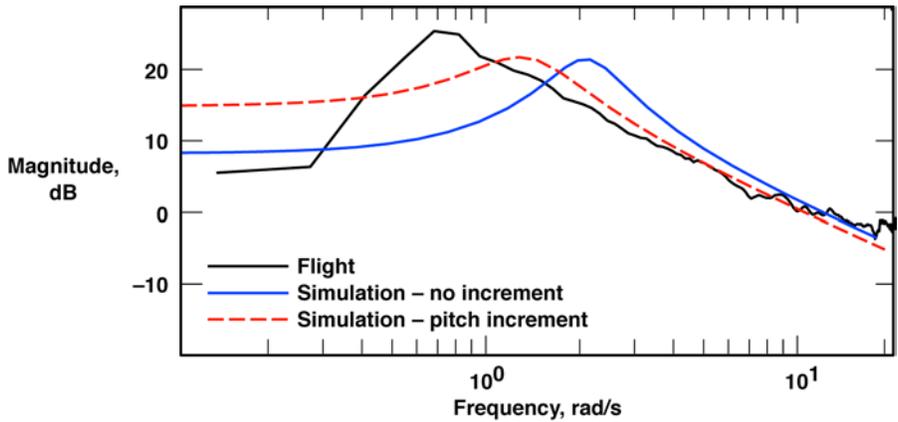


Figure 7. Bode plot comparison of flight data to simulation data – pitch rate to stabilator deflection.

Another simulation comparison was run in an effort to determine if changes to the baseline aerodynamics determined by the PID in flight would have resulted in better handling qualities had the system been flown in a closed-loop fashion. Handling qualities ratings were determined using the standard Cooper-Harper rating scale. Handling qualities analysis run on the frequency sweep flight data using the lower order equivalent system (LOES) method⁷ predicts that the handling qualities will lie in the Level 2 range. Evaluating an automated pitch frequency sweep performed in the simulation gives a similar prediction for the control anticipation parameter (CAP) – Figure 8 shows these two predictions as the blue (flight data) and red (simulation with original aerodynamics) asterisks. As each new pitch derivative increment was incorporated into the nonlinear simulation, the CAP moved into the Level 1 area. Although the CAP moved into the Level 1 boundary, the CAP value is still different from the desired or reference value. Differences observed here can possibly be attributed to the use of the tuned-down PID, where PTNN errors in the nonestimated derivatives may still be present.

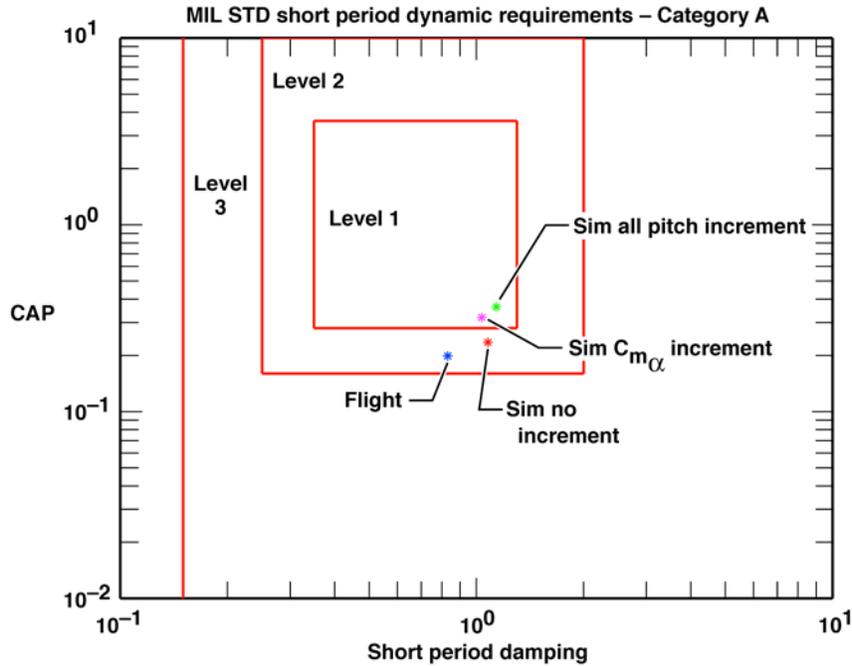


Figure 8. CAP comparisons for pitch aerodynamic derivative increments.

Another method that predicts Cooper-Harper ratings was also used with the intent of showing additional evidence of an improvement in pitch handling qualities. A determination of the average Cooper-Harper (ACH) rating⁸ was made for each instance using the Smith-Geddes criteria. The ACH rating improved (decreased) by more than 2 when the pitch derivatives identified in flight by the PID were added to the baseline derivatives being sent to the SOFFT controller. Figure 9 shows these improvements. This figure supports the conclusion that had the Gen 1 system been run closed-loop in real time, better pitch handling qualities of the airplane would have been observed during flight.

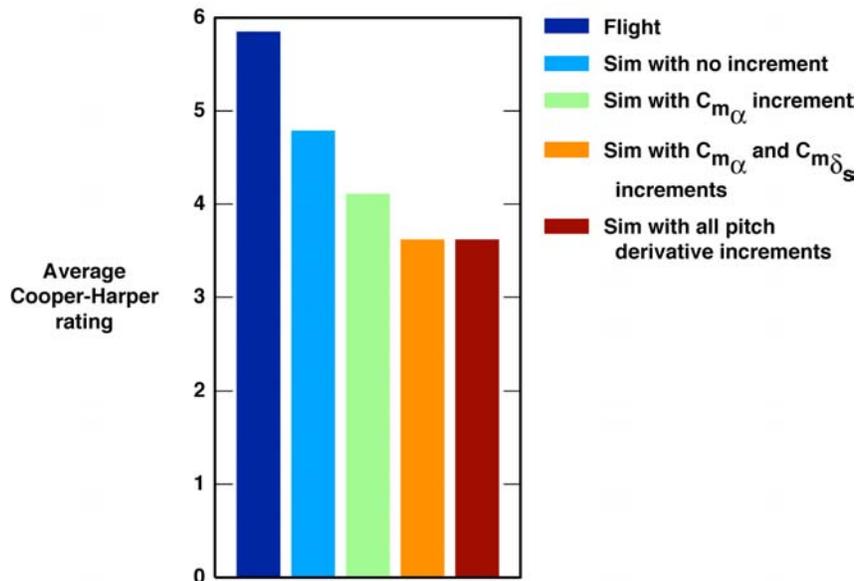


Figure 9. Comparison of average Cooper-Harper ratings for pitch derivatives – flight data as opposed to nonlinear simulation data for pitch frequency sweep maneuver.

V. Summary

This report presented flight-test results for an adaptive controller using an online learning neural network. A dynamic cell structure neural network was used in conjunction with a real-time parameter identification algorithm to estimate aerodynamic stability and control derivative increments to the baseline aerodynamic derivatives. This set of open-loop flight tests was done in preparation for a future phase of flights in which the learning neural network and parameter identification algorithm output would be providing the controller with aerodynamic stability and control derivative updates in real time. Two flight maneuvers were analyzed – a pitch frequency sweep and an automated flight-test maneuver. An examination of the flight data recorded showed that neural network operating in combination with the parameter identification algorithm identified a change to the baseline aerodynamic derivatives in real time during flight.

Handling qualities improvement was shown by generating frequency responses using a nonlinear simulation updated with the flight-identified increments. Analysis of the control anticipation parameter and the average Cooper-Harper rating from the Smith-Geddes criteria both showed a handling qualities improvement from Level 2 to Level 1 when updated with the flight-identified aerodynamic pitch derivative values.

References

¹Smith, Mark S., Moes, Timothy R., and Morelli, Eugene A., Real-Time Stability and Control Derivative Extraction from F-15 Flight Data, NASA/TM-2003-212027, September 2003.

²Jorgensen, Charles C., Direct Adaptive Aircraft Control Using Dynamic Cell Structure Neural Networks, NASA TM 112198, May 1997.

³Smolka, James W., et al., “F-15 ACTIVE Flight Research Program,” *1996 Society of Experimental Test Pilots 40th Symposium Proceedings*, Paper 1854, September 1996.

⁴Urnes Sr., James, Davidson, Ron, and Jacobson, Steve, “A Damage Adaptive Flight Control System Using Neural Network Technology,” American Control Conference Paper, June 2001.

⁵Hageman, Jacob, Smith, Mark, and Stachowiak, Susan, “Integration of Online Parameter Identification and Neural Network for In-flight Adaptive Control,” AIAA Paper 2003-5700, August 2003.

⁶Moes, Timothy R., Smith, Mark S., and Morelli, Eugene A., “Flight Investigation of Prescribed Simultaneous Independent Surface Excitations for Real-Time Parameter Identification,” AIAA Paper 2003-5702, August 2003.

⁷Hodgkinson, J., and LaManna, W. J., “Equivalent System Approaches to Handling Qualities Analysis and Design Problems of Augmented Aircraft,” AIAA Paper 77-1122, August 1977.

⁸Smith, Ralph H. and Geddes, Norman D., “Handling Quality Requirements for Advanced Aircraft Design: Longitudinal Mode,” Air Force Flight Dynamics Laboratory Paper AFFDL-TR-78-154, August 1979.